

哈 尔 滨 理 工 大 学

课 程 设 计

题 目： 基于 TCP的 CS模式程序设计

院 系： 计算机科学与技术学院网络工程系

班 级： 网络 11-2 班

学 号：

姓 名：

同 组 成 员：

指 导 教 师：

成 绩：

2014 年 06 月 27 日

一. 系统设计的目标

1.1 选题目的和意义

在 Windows 操作系统下, 使用 Socket 编程实现两个进程之间的通信, 两个进程间使用 TCP 建立可靠连接, 熟悉 TCP 协议以及套接字编程原理; 两个进程间使用 C/S 工作模式, 在此结合并发服务器原理, 实现服务器同时与多进程之间进行通信, 深入了解 C/S 工作模式。

TCP 是传输层的应用协议, 可通过多方面来保证数据传输的可靠性, 它采用了停止等待方式来回应数据的接收, 只有当接收到数据时, 用户才发送数据确认报文, 同时此协议使用滑动窗口方法来控制网络拥塞, 提高数据传输的可靠性。

基于 TCP/IP 协议和套接字编程原理为服务器端完成基本数据配置, 主要采用套接字编程原理, 通过调试运行并且经过完善后, 实现并发服务器的功能。在这次课程设计的过程中, 对于网络编程有了更深一步的了解。首先是对于计算机网络基础这方面, 对于网络体系中的几种分层方法, 每层的作用都由于系统的理解, 特别是对于 Socket 编程这一块, 套接字的创建、绑定以及使用, 基于 TCP、UDP 的编程以及并发服务器的原理都有了深一步的理解, 这次课程设计对于以后进行网络的开发有很大的帮助。

即时通信(Instant Message) , 由于其具有实时性、跨平台性、成本低、效率高等优点而受到广泛的使用。设计并实现一个能够处理

多用户进行实时、安全的即时通信系统具有较强的现实意义。即时通信的底层通信是通过 **SOCKET** 套接字接口实现的。当前的主流 **UNIX** 系统和微软的 **WINDOWS** 系统都在内核提供了对 **SOCKET** 套接字接口的支持。使用这个统一的接口，可以编写一个可移植的 **TCP/IP** 通信程序。使信息能够在 **INTERNET** 上可靠的传输。 本文设计并实现了基于局域网内的简单即时通信系统，系统采用 **C/S** 模式，底层通信通过 **SOCKET** 套接字接口实现，服务器负责客户端的登录验证，好友信息的保存和心跳报文的发送。客户端采用 **P2P** 方式实现消息传递，并能实现文件的传输。本文首先讨论了同步套接字，异步套接字，多线程并发执行任务等；然后阐述了客户端、服务器如何使用 **XMI** 序列化的消息进行通信。

本课题目标是实现局域网用户之间的即时交流和文件传输，通过基础的网络 **SOCKET** 编程，为局域网内部的即时通信提供一个简单而较安全的解决方案。

1.2 题目开发背景

即时通信是一个终端连往一个即时通信网路的服务。即时通信不同于 **e-mail** 在于它的交谈是实时的。大部分的即时通信服务提供了 **presence awareness** 的特性——显示联络人名单，联络人是否在线与能否与联络人交谈。

最早的即时通信软件是 **ICQ**，**ICQ** 是英文中 **I seek you** 的谐音，意思是我找你。四名以色列青年于 1996 年 7 月成立 **Mirabilis** 公司，

并在 11 月份发布了最初的 ICQ 版本，在六个月内有 85 万用户注册使用。

在因特网上受欢迎的即时通信服务包含了 MSN Messenger、AOL Instant Messenger、Yahoo! Messenger、NET Messenger Service、Jabber、ICQ 与 QQ 这些服务有赖于许多想法更久的（与普遍）的在线聊天媒介，如 Internet Relay Chat 一样知名。

1970 年代早期，一种更早的即时通信形式是柏拉图系统 (PLATO system)。之后在 1980 年代，UNIX/Linux 的交谈实时信息被广泛的使用于工程师与学术界，1990 年代即时通信更跨越了因特网交流。1996 年 11 月，ICQ 是首个广泛被非 UNIX/Linux 使用者用于因特网的即时通信软件。在 ICQ 的介绍之后，同时在许多地方有一定数量的即时通信方式发展，且各式的即时通信程序有独立的协议，无法彼此互通。这引导使用者同时执行两个以上的即时通信软件，或者他们可以使用支持多协议的终端软件，如 Gaim、Trillian 或 Jabber。

近年来，许多即时通信服务开始提供视讯会议的功能，网络电话 (VoIP)，与网络会议服务开始整合为兼有影像会议与实时信息的功能。于是，这些媒体的分别变的越来越模糊。

1.2 国内外研究现状

1.2.1 国外研究现状

当今，国际上对网络通信系统研究的较好的公司有，思科，Sun，Ms等公司，思科主要研究的是底层的传输；MS Sun 公司研究的是应用层。其中 ms 公司凭借其在操作系统的垄断地位，为了在网络的发展中取得先机，采用了各种各样的手段。但是，其捆绑的 msn，无论从功能上，还是技术上来说，都不算是非常先进的。当然，ie，同样也不是很受人青睐，这让人想起了，当年的网景公司，网景只是生不逢时。MS不择手段的想打跨网景，可见其对网络的重视。如今，Sun 公司在网络应用上捷足先登，凭借着 Java，Sun 在网络的应用上领先于 MS 微软，想用同样的办法搞跨对手，因此它拿出了 Visual c#，来对抗 Java。这些都是应用层面的开发工具。应用层上的产品就更显种类繁多。ICQ几乎是国际上通用的即时通信工具，由于在我国它的应用不是很广，所以，其原理也很少被介绍。msn，是 MS的产品，同样在国内没什么市场，所以，对其原理，也很少被讨论过。至于 ie，是在 Visual c++下开发的产品，虽然有严重的安全隐患，不过，至少能在某种程度上代表当今国际研究的水平。此外，国际上最近出先了新的浏览器 Firefox，其性能据说是远高于 ie，也许在网络的天，Ms又有了更强劲的对手。

1.2.2 国内研究现状

国内在应用层上的网络应用软件目前发展异常的火爆，因为我国有着网络应用的最大的市场，现在国内网络的基础性建设发展迅速，应用软件也层出不穷，其中，在游戏的领域中，网络通信的工作做的不

错，如联众游戏平台，还有其他的一些平台，这些平台基本上都是基于 VC+的，用的都是 Socket 通信，但是为了效率，这些平台没有用 MFC提供的 CSocket 类，而是直接用 Socket 进行通信。所以效率上不错。此外，tencent 的即时通信，也是做的很好的，从某中程度上来说，代表了国内最高的水平。

1.3 TCP/IP 体系结构

TCP/IP 的中文译名为传输控制协议/网际协议，又叫网络通讯协议，这个协议是 Internet 最基本的协议，Internet 国际互联网络的基础，简单地说，就是由网络层的 IP 协议和传输层的 TCP协议组成的。

TCP/IP 协议并不完全符合 OSI 的七层参考模型。传统的开放式系统互连参考模型，是一种通信协议的七层抽象的参考模型，其中每一层执行某一特定任务。该模型的目的是使各种硬件在相同的层次上互相通信。这七层是：物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。而 TCP/IP 通讯协议采用了四层的层级结构，每一层都呼叫它的下一层所提供的网络来完成自己的需求，如下图所示。

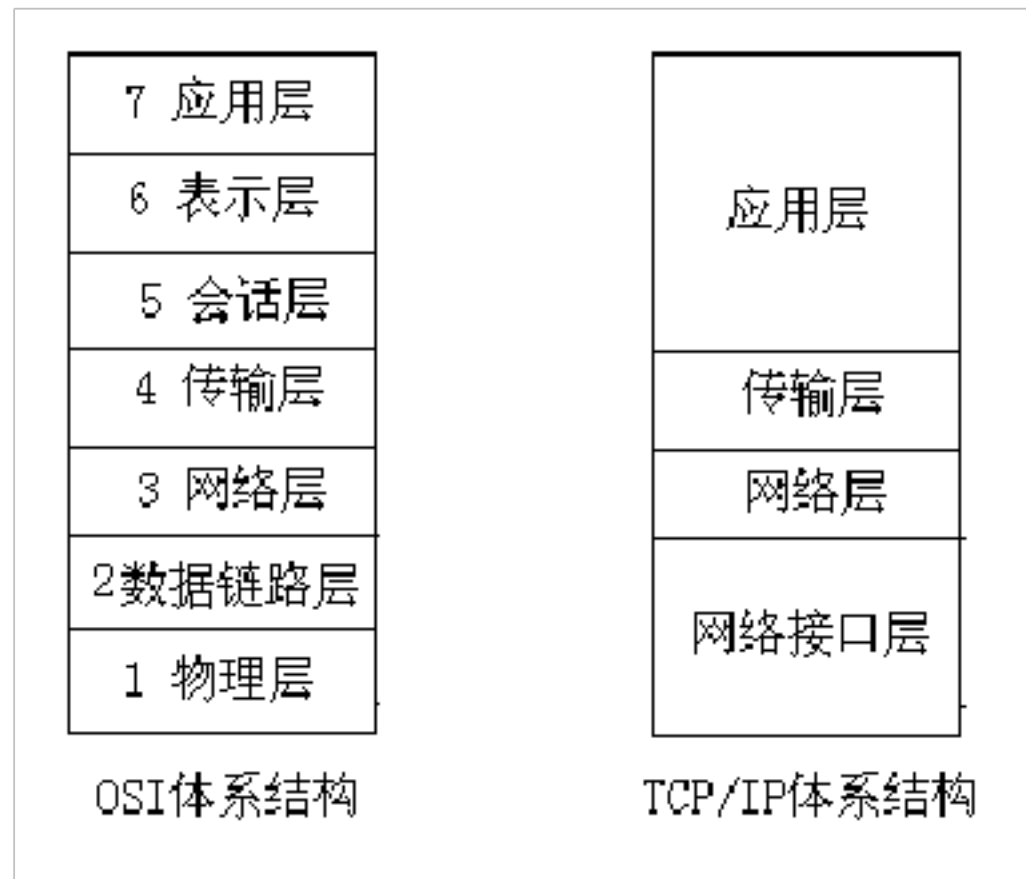


图 1 计算机网络体系结构

应用层：应用程序间沟通的层，如简单电子邮件传输（SMTP），文件传输协议（FTP），网络远程访问协议（Telnet）等。

传输层：在此层中，它提供了节点间的数据传送，应用程序之间的通信服务，主要功能是数据格式化，数据确认和丢失、重传等。如传输控制协议（TCP），用户数据报协议（UDP）等，TCP和UDP给数据包加入传输数据并把它传输到下一层中，这一层负责传送数据，并且确定数据已被传送并到达接收。

网络层：负责提供基本的数据封包传送功能，让每一块数据包都能够到达目的主机（但不检查是否正确而接收），如网际协议（IP）。

网络接口层：接收IP数据包并进行传输，从网络上接收物理帧，抽取IP数据报转交给上一层，对实际的网络媒体的管理，定义如何使用实际网络（如Ethernet）来传送数据。

1.4 TCP 传输



图 2 TCP 连接中分组交换以及客户端与服务器的状态转换

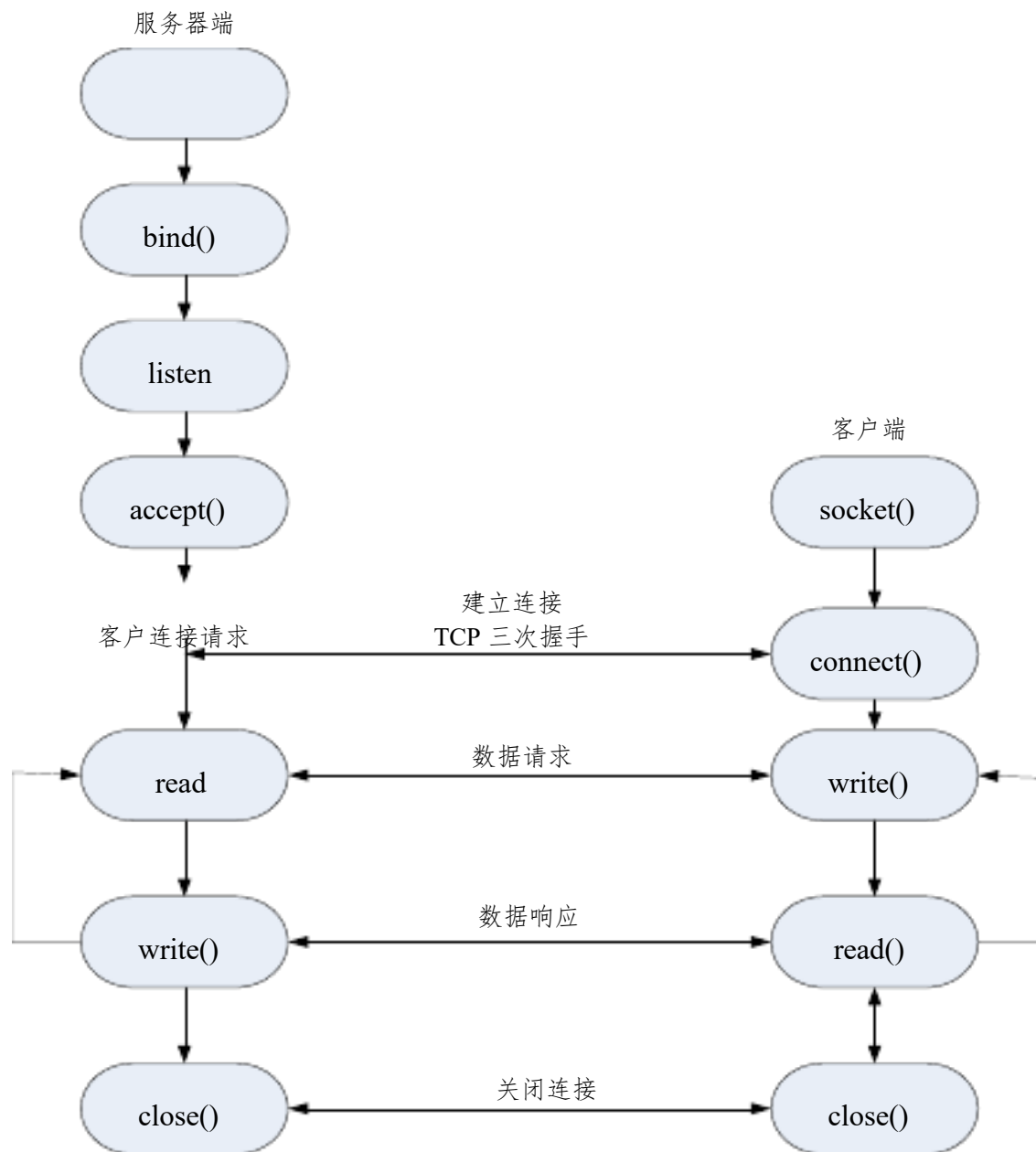
TCP协议在 IP 协议之上，提供可靠的传输服务。这种服务的特点：可靠、全双工、流式和无结构传输。TCP协议使用了积极确认和重传的技术来实现可靠传输。接收者在收到发送者发送的数据后，必须发送一个相应的确认（ACK）消息，表示他已经收到数据。发送者保存发送的数据的记录，在发送下一个数据之前等待这个数据的确认消息在他发送这个数据的同时，启动了一个计时器。在一定时间内没有接收到确认消息，就认为这个数据在传输过程中丢失了，接着就会发送这个数据。同时 TCP实体又采用了滑动窗口协议。当发送方传送一个数据报到达目的地时，接收方的 TCP 实体向发送方发送一个数据报，其中包含一个确认序号，它等于希望收到的下一个数据报的顺序

号。建立了一个 TCP 连接，需要三次握手过程，连接才能成功建立，如图二所示。

握手过程的第一段的代码位设置为 SYN，序号为 J，表示开始一次握手。接收方收到此报文段后，向发送方回发一个报文段，代码位为 SYN 和 ACK，序号为 K，确认序号为 J+1，表示接收方同意建立此连接。发送者收到此报文段后，发送 ACK 建立连接，完成三次握手以后，双方就开始正式传送数据。在通信完成阶段，两方通过四次断开来释放连接，如图二，客户端请求释放连接，发送有效位为 FIN，服务器端给一个确认信号，若服务器端没有数据发送，便向客户端发送 FIN 请求，客户端对此回应，等待 2MSL 彻底断开连接。

1.5 C/S 模式

鉴于此次设计目的，采用 Linux 环境下使用 TCP 实现点到点的可靠连接，实现基于 TCP/IP 协议的面向连接的通信，连接双方分为客户端和服务端，主要实现过程如图 3 所示。



3 TCP 客户/服务器的套接字连接

TCP套接字编程中，服务器实现的步骤：

- (1) 使用 `socket()` 函数创建套接字。
- (2) 将创建的套接字绑定到指定的地址结构。
- (3) `Listen()` 函数设置套接字为监听模式，使服务器进入被动打开的状态。
- (4) 接受客户端的连接请求，建立连接。
- (5) 接收、应答客户端的数据请求。
- (6) 终止连接。

客户端实现的步骤：

- (1) 使用 `socket()` 函数创建套接字。

- 2) 用 `connect()` 函数建立一个与 TCP 服务器的连接。
- (3) 发送数据请求, 接收服务器的数据应答。
- (4) 终止连接。

综上所述, TCP/IP 应用中的服务器必须先于客户端进程启动, 直到对客户端的响应结束或强迫终止。

. 系统原理

开发配置环境简介

CodeBlocks 安装，和普通安装一样，你不会选择可以一直 next 完成，不做详细叙述。注意：当中有一步让你选择编译器，你可以选择 GUN GCC Compiler，并设置位默认（default）。如果没做也没事儿，可以在编译器（Compiler）设置进行设置。

GTK+ 配置 安装 GTK 解压
 gtk+bundle_2.24.10-20120208_win32.zip 到 。也可以是别处，你个人喜好，随便哪儿都行。配置环境变量 添加
 “ ” 到系统环境变量(path) 注意：GTK解压的位置不同，你相应变一下就行了。

导出编译库信息 导出编译库信息，一会儿要在 Code::Blocks

上配置编译、链接环境。 `pkg-config --cflags gtk+-2.0 >`

`cflags.txt cflags.txt pkg-config --libs gtk+-2.0 >`

`libs.txt libs.txt` 还是那句话，安装位置不同，内容不同，

不要直接从我这里拷贝。

配置 Code::Blocks 编译链接环境 菜单上依次点击：

Settings--Compiler settings 会出现 Compiler settings

设置窗口： a. 此处把 `cflags.txt` 内容复制至文本框

2.2 开发工具及技术简介

DevC:Dev-C++是一个 C&C++开发工具，它是一款自由软件，遵守 GPL 协议。它集合了 GCC MinGW等众多自由软件，并且可以取得最新版

作，并且你拥有对这一切工具自由使用的权利，包括取得源代码等，前提是你也必须遵守 GNU 协议。Dev-C++ 每一天都在进步着，因为它是一个自由软件。Dev-C++ 是一个非常实用的编程软件，多款著名软件均由它编写而成，它在 C 的基础上，增强了逻辑性。

发展 C++ 是从 C 语言中发展而来的。C 语言是 1972 年由美国贝尔实验室 (AT&TBell) 的 D.M.Ritchie 研制成功的。它不是为了初学者设计的，而是为计算机专业人员设计的。最初它是作为写 UNIX 操作系统的一种工具，在贝尔实验室内部使用。后来 C 语言不断改进，人们发现它功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好，既具有高级语言的优点，又具有低级语言的许多优点，特别适合于写系统软件，因此 C 语言从实验室走向美国，从美国走向世界。到 20 世纪 70 年代，它已风靡全世界。无论是在中国还是在外国，C 语言都成为了计算机开发人员的基本功。

为了解决软件设计的危机，在 20 世纪 80 年代，人们提出了面向对象的程序设计 (object oriented programming, OOP)，需要设计出能支持面向对象的程序设计方法的新的语言。在实践中，人们发现由于 C 语言是如此的深入人心，使用如此广泛，面对程序设计方法的革命，最好的办法不是另外发明一种语言去代替它，而是在它原有的基础上加以发展。在这种形式下，C++ 应运而生。C++ 是由贝尔实验室 (AT&TBell) 的 Bjarne Stroustrup 博士及其同事于 20 世纪 80 年代初在 C 语言的基础上开发成功的。

AT&TBell 发布的第一个 Dev-C++ 编译系统实际上是一个预编译器(前端编译器),真正的 Dev-C++ 程序是在 1988 年诞生的。应用

Dev-C++ 目前是 NOI NOIP 等比赛的指定工具,唯一的缺点就是 Debug 功能较弱。

Codeblocks: 它是基于 wxWidget 库的程序。首先,因为它就是用 C++ 开发的,所以在速度上明显比 Eclipse 快。其次,因为是一个专门的 C++ IDE 所以在配置方面也非常符合 C++ 开发的习惯,功能几乎可以和 VC2005/2007 媲美。最后,它还是一个开源的 IDE,并且它的体积比 Eclipse 和 VC 小了不止一个数量级——整个压缩后的源码包只有 6M 多!其实说实话,我也是刚开始用 Code::Blocks,但是我敢负责任的说,它真的很好。以前编程和调试都一直在 VIM Emacs 和 gdb 里挣扎,命令方式的 GDB 调试把握折腾的都崩溃了。突然接触了 Code::Blocks 以后,马上让我找到了当前用 VC 编程和调试的快乐的感觉。就是前几天,我在 GDB 下调试一个 C++ 程序时的不便,尝试了 DDD 还是觉得不方便,于是偶然间试了一下 Code::Blocks,一会的时间问题就解决了。虽然它使用的还是 GCC+GDB 但是它对其包装以后使我使用起来感到了极大的方便

2.3 相关协议及结构介绍

套接字 套接字这个术语并没有定义某个协议:它具有两层含义,但两者都与一个协议相关。第一个含义是套接字编程 API,它最初由

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/195201144040012013>